

Package: wbCorr (via r-universe)

June 9, 2026

Title Bivariate Within- and Between-Cluster Correlations

Version 0.3.1

Date 2026-05-23

Description Separates supplied variables into within- and between-cluster components and calculates bivariate correlations for each level separately. The centered-score decomposition corresponds to commonly used between- and within-cluster correlations discussed by Tu et al. (2025) <[doi:10.1002/sim.10326](https://doi.org/10.1002/sim.10326)>. The package is also motivated by the distinction between within- and between-person variation described by Curran and Bauer (2011) <[doi:10.1146/annurev.psych.093008.100356](https://doi.org/10.1146/annurev.psych.093008.100356)> and by Hamaker (2024) <[doi:10.1080/00273171.2022.2155930](https://doi.org/10.1080/00273171.2022.2155930)>. The package is intended for longitudinal or otherwise clustered data where researchers need transparent correlation matrices before fitting more complex multilevel models.

License MIT + file LICENSE

Encoding UTF-8

Suggests htrr, spelling, testthat (>= 3.0.0)

Imports methods, writexl

Config/testthat/edition 3

Language en-US

URL <https://github.com/Pascal-Kueng/wbCorr>

BugReports <https://github.com/Pascal-Kueng/wbCorr/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Pascal Küng [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-7346-9414>>)

Maintainer Pascal Küng <pascal.kueng@psychologie.uzh.ch>

Depends R (>= 3.5.0)

Repository <https://cran.r-universe.dev>
Date/Publication 2026-06-09 16:00:02 UTC
RemoteUrl <https://github.com/cran/wbCorr>
RemoteRef HEAD
RemoteSha 6d0c76cda7ed731196f811dd1bd7d49f9a97c17d

Contents

get_ICC	2
get_matrix	3
get_table	4
plot,wbCorr-method	5
print,wbCorr-method	7
show,wbCorr-method	8
simdat_intensive_longitudinal	8
to_excel	9
update_wbCorr	10
wbCorr	11
wbCorr-class	14

Index **16**

get_ICC	<i>Return all ICCs for the original variables.</i>
---------	--

Description

You can use `get_ICC()` or `get_ICCs()` interchangeably.

Usage

```
get_ICC(object)
get_ICCs(object)
get_icc(object)
```

Arguments

`object` A `wbCorr` object, created by the `wbCorr()` function.

Value

A dataframe with ICCs for all variables. ICC is obtained by fitting mixed effects models and extracting the variance components. Then, the formula between- variance / total- variance is applied.

See Also[wbCorr](#)**Examples**

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns the ICCs:
ICCs <- get_ICC(correlations)
print(ICCs)
```

`get_matrix`*Return matrices for within- and/or between-cluster correlations.*

Description

You can use `summary()`, `get_matrices()`, or `get_matrix()` interchangeably. Merged matrices include the ICC on the diagonal. For more detailed statistics, use `get_table()`.

Usage

```
get_matrix(object, which = c("within", "between", "merge"), ...)

get_matrices(object, which = c("within", "between", "merge"), ...)

## S4 method for signature 'wbCorr'
summary(object, which = c("within", "between", "merge"), ...)
```

Arguments

<code>object</code>	A <code>wbCorr</code> object, created by the <code>wbCorr()</code> function.
<code>which</code>	A string or a character vector indicating which summaries to return. Options are 'within' or 'w', 'between' or 'b', and various merge options like 'merge', 'm', 'merge_wb', 'wb', 'merge_bw', 'bw'. Default is <code>c('within', 'between', 'merge')</code> .
<code>...</code>	Additional arguments passed to the base summary method

Value

A list containing the selected matrices of within- and/or between-cluster correlations, and ICCs on the diagonals for merged matrices.

See Also

[get_tables](#), [wbCorr](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns a correlation matrix with stars for p-values:
matrices <- summary(correlations) # the get_matrix() and get_matrices() functions are equivalent
print(matrices)

# Access specific matrices by:
# Option 1:
matrices$within
# Option 2:
within_matrix <- summary(correlations, which = 'w') # or use 'within'
merged_within_between <- summary(correlations, which = 'wb')
print(within_matrix) # could be saved to an excel or csv file (e.g., write.csv)
```

get_table

Retrieve full tables for both within- and/or between-cluster correlations for a wbCorr object.

Description

This function has an alias `get_tables()` which can be used interchangeably. For correlations matrices, see the `summary()` function.

Usage

```
get_table(object, which = c("within", "between"))

get_tables(object, which = c("within", "between"))
```

Arguments

<code>object</code>	A <code>wbCorr</code> object, created by the <code>wbCorr()</code> function.
<code>which</code>	A character vector indicating which correlation table to return. Options are 'within' or 'w', and 'between' or 'b'.

Value

A list containing the selected tables of within- and/or between-cluster correlations.

See Also

[summary, wbCorr](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# returns a list with full detailed tables of the correlations:
tables <- get_table(correlations) # the get_tables() function is equivalent
print(tables)

# Access specific tables by:
# Option 1:
tables$between
# Option 2:
within_table <- get_tables(correlations, which = 'w') # or use 'within' or 'between'
print(within_table) # within_table could be saved to an excel or csv file (e.g., write.csv)
```

plot,wbCorr-method *Plot within- and between associations*

Description

Plots the centered variables of the provided dataframe against each other. Choose whether to plot the between-centered variables (representing the between-cluster correlations by plotting cluster means) or the within-centered variables (representing the within-cluster correlations by plotting deviations from person-means). A regression line is provided and the corresponding coefficient with significance displayed.

Usage

```
## S4 method for signature 'wbCorr'
plot(
  x,
  y,
  which = NULL,
  plot_NA = TRUE,
  standardize = TRUE,
  outlier_detection = "zscore",
  outlier_threshold = "recommended",
  type = "p",
  pch = 20,
```

```

    dot_lwd = 2,
    reg_lwd = 2,
    ...
)

```

Arguments

<code>x</code>	A <code>wbCorr</code> object to be plotted.
<code>y</code>	Choose which correlations to plot ('within' / 'w' or 'between' / 'b'); can be used as a positional argument.
<code>which</code>	Can be used as an alternative to 'y' (e.g., <code>which = 'w'</code>). It has the same functionality as 'y', but takes precedence if both are specified.
<code>plot_NA</code>	Boolean. Whether variables that have no variation on the selected level should be plotted or not.
<code>standardize</code>	Boolean. Whether the dataset should be standardized. If TRUE, the regression coefficient is equivalent to the pearson correlation.
<code>outlier_detection</code>	If FALSE, outliers will not be marked in red. Otherwise you may provide the method. Choose from: 'zscore', 'mad', or 'tukey'.
<code>outlier_threshold</code>	If 'recommended', the threshold for 'zscore' and 'mad' will be set to 3, and for 'tukey' to 1.5. You can provide and other numeric here.
<code>type</code>	points, lines, etc. see <code>?base::plot</code> for available types).
<code>pch</code>	Graphical parameter. Select which type of points should be plotted.
<code>dot_lwd</code>	Graphical parameter. Set size of the points.
<code>reg_lwd</code>	Graphical parameter. Set thickness of the regression line.
<code>...</code>	further options to be passed to the base plot (<code>pairs</code>) function.

Value

Invisibly returns the supplied `wbCorr` object. Called for the side effect of drawing a pairs plot of the selected within- or between-cluster centered variables.

See Also

[wbCorr](#)

print,wbCorr-method *Print Method for the wbCorr Class*

Description

Prints a summary of the wbCorr object.

Usage

```
## S4 method for signature 'wbCorr'  
print(x, ...)
```

Arguments

x	A wbCorr object.
...	Additional arguments, currently unused.

Value

Invisibly returns the supplied wbCorr object. Called for the side effect of printing a compact summary of the within-cluster table, between-cluster table, and ICC table.

See Also

[wbCorr](#)

Examples

```
# Example  
data("simdat_intensive_longitudinal")  
correlations <- wbCorr(simdat_intensive_longitudinal,  
                      cluster = 'participantID',  
                      confidence_level = 0.95,  
                      method = 'spearman',  
                      weighted_between_statistics = FALSE)  
print(correlations)
```

show, wbCorr-method *Show Method for the wbCorr Class*

Description

Shows a summary of the wbCorr object, equivalent to the print method.

Usage

```
## S4 method for signature 'wbCorr'  
show(object)
```

Arguments

object A wbCorr object.

Value

Invisibly returns the supplied wbCorr object. Called for the side effect of showing the same compact summary as print().

See Also

[wbCorr](#), [print.wbCorr](#)

Examples

```
# Example using the iris dataset  
cors <- wbCorr(iris, iris$Species, weighted_between_statistics = TRUE)  
show(cors)
```

simdat_intensive_longitudinal
Simulated Intensive Longitudinal Dataset

Description

A simulated intensive longitudinal dataset to test the package capabilities. This dataset contains 80 participants, day, and three variables (var1, var2, and var3) that are all correlated on both within- and between-levels.

Format

A data frame with the following columns:

- participantID** Identifier for each participant (integer)
- day** Day variable varying only within-person (integer)
- var1** Variable 1 (numerical)
- var2** Variable 2 (numerical)
- var3** Variable 3 (numerical)

Details

The within-person correlations are all positive:

- var1 & var2: 0.1
- var1 & var3: 0.3
- var2 & var3: 0.8

The between-person correlations are all negative:

- var1 & var2: -0.5
- var1 & var3: -0.4
- var2 & var3: -0.2

Time trends (within):

- var1 & time: 0.0
- var2 & time: 0.0
- var3 & time: 0.4

Source

A simulated dataset by P. Küng

to_excel

Saves the passed summary or table to excel

Description

Use `to_excel(get_matrix(wbCorrObject))` or `to_excel(get_table(wbCorrObject))` to save the provided table/matrix to an excel file.

Usage

```
to_excel(SummaryObject, path = file.path(getwd(), "wbCorr.xlsx"))
```

Arguments

SummaryObject A summary or matrix object, such as those returned by `get_matrix()` or `get_table()`.
 path Specify the filename and a path. If no path is provided, the file will be saved to the current working directory.

Value

Writes an Excel file (.xlsx) to disk.

See Also

[get_tables](#), [wbCorr](#), [get_matrix](#)

Examples

```
# Importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# Create object:
correlations <- wbCorr(data = simdat_intensive_longitudinal,
                      cluster = 'participantID')

# Returns a correlation matrix with stars for p-values:
matrices <- get_matrix(correlations) # summary(correlations) works too.

to_excel(matrices, path = tempfile(fileext = ".xlsx"))
```

update_wbCorr

Check for updates of wbCorr

Description

This function checks if there is a newer version on GitHub by comparing the version numbers in the local and remote DESCRIPTION files. It only runs when called explicitly by the user and does not install updates.

Usage

```
update_wbCorr(ask = FALSE)
```

Arguments

ask Deprecated and ignored.

Value

An integer: 1 if there's a newer version available, 0 if the current version is the latest, or NULL if there was an error accessing the remote DESCRIPTION file.

 wbCorr

wbCorr

Description

The `wbCorr` function creates a `wbCorr` object containing within- and between-cluster correlations, p-values, and confidence intervals for a given dataset and clustering variable. The object can be plotted.

Usage

```
wbCorr(
  data,
  cluster,
  confidence_level = 0.95,
  method = "pearson",
  bootstrap = FALSE,
  nboot = 1000,
  inference = c("analytic", "none", "cluster_bootstrap"),
  weighted_between_statistics = NULL,
  between_weighting = c("equal_clusters", "cluster_size"),
  between_inference = c("analytic", "none"),
  centering_rows = c("pairwise_complete", "all_available")
)
```

```
wbcorr(
  data,
  cluster,
  confidence_level = 0.95,
  method = "pearson",
  bootstrap = FALSE,
  nboot = 1000,
  inference = c("analytic", "none", "cluster_bootstrap"),
  weighted_between_statistics = NULL,
  between_weighting = c("equal_clusters", "cluster_size"),
  between_inference = c("analytic", "none"),
  centering_rows = c("pairwise_complete", "all_available")
)
```

Arguments

<code>data</code>	A dataframe containing numeric variables for which correlations will be calculated.
<code>cluster</code>	A vector representing the clustering variable or a string with the name of the column in <code>data</code> that contains the clustering variable.

confidence_level	A numeric value between 0 and 1 representing the desired level of confidence for confidence intervals (default: 0.95).
method	A string indicating the correlation method to be used. Supported methods are 'pearson', 'spearman', and 'spearman-jackknife'. (default: 'pearson'). 'pearson': Pearson correlation method uses t-statistics to determine confidence intervals and p-values. 'spearman': Spearman correlation method uses the Fisher z-transformation for confidence intervals and p-values. 'spearman-jackknife': Employs the Euclidean jackknife technique to compute confidence intervals, providing more robust confidence intervals in the presence of non-normal data or outliers. Note that p-values are not available when this method is selected.
bootstrap	Deprecated logical alias for inference = "cluster_bootstrap".
nboot	Specifies the amount of bootstrap samples (default: 1000).
inference	A string specifying how p-values and confidence intervals are calculated. "analytic" uses the usual correlation-test approximation. "none" returns coefficients without p-values or confidence intervals. "cluster_bootstrap" resamples top-level clusters with replacement and recomputes the full decomposition in each bootstrap sample.
weighted_between_statistics	Deprecated logical alias for between_weighting. If TRUE, between_weighting = "cluster_size"; if FALSE, between_weighting = "equal_clusters".
between_weighting	A string specifying the between-cluster estimand. "equal_clusters" correlates pair-specific cluster means with each cluster contributing equally. "cluster_size" computes a sample-size weighted correlation of pair-specific cluster means, using the number of complete observation pairs in each cluster as weights.
between_inference	A string specifying whether between-cluster p-values and confidence intervals are calculated analytically ("analytic") or omitted ("none"). Analytic inference for "cluster_size" weighted between correlations uses k - 2 cluster-level degrees of freedom for Pearson correlations and is approximate. Ignored when inference = "none" or inference = "cluster_bootstrap".
centering_rows	A string specifying which rows are used to estimate cluster means for within- and between-cluster decomposition. "pairwise_complete" uses only rows where both variables in the current pair are observed. "all_available" estimates each variable's cluster mean from all available rows for that variable, then correlates the pair on complete rows.

Details

Calculates bivariate within- and between-cluster correlations for clustered data, such as repeated measures nested in persons, dyads, teams, or other groups. Only recommended for continuous or binary variables.

For every variable pair, correlations are computed on rows where both variables and the cluster variable are observed. By default, centering_rows = "pairwise_complete" also estimates cluster means from this same complete-pair row set. This keeps the within residuals centered for the

actual pairwise sample and makes the between correlation a correlation of matched pair-specific cluster means.

With `centering_rows = "all_available"`, each variable's cluster mean is estimated from all available rows for that variable before the pairwise correlation is computed. This can make the cluster means more stable when data are missing. It also mirrors a common multilevel-model preprocessing workflow, where person means are often created before the model applies complete-case filtering. That workflow is defensible in multilevel models. In `wbCorr`, however, the variables are treated symmetrically as a descriptive bivariate decomposition, so all-available centering means the two cluster means in a pair may be based on different occasions. For that reason, `"pairwise_complete"` is the default.

The within-cluster correlation is the pooled residual correlation. For a given pair, each observed value is centered around its cluster mean for that same complete-pair row set, and the correlation is computed on the resulting residuals. For Pearson within-cluster correlations, analytic inference uses $N_{\text{pair}} - k_{\text{pair}} - 1$ degrees of freedom, where N_{pair} is the number of complete observation pairs and k_{pair} is the number of clusters contributing at least one complete pair. This analytic test is a working approximation because residual pairs can still be dependent within clusters; for publication-level inference in intensive longitudinal data, prefer `inference = "cluster_bootstrap"`.

The between-cluster correlation is computed from pair-specific cluster means. With `between_weighting = "equal_clusters"`, every cluster contributes one equally weighted mean. With `between_weighting = "cluster_size"`, cluster means are weighted by the number of complete observation pairs in that cluster. Analytic p-values and confidence intervals for cluster-size weighted between correlations are approximate; use `between_inference = "none"` to report only the weighted coefficient.

With `inference = "cluster_bootstrap"`, `wbCorr` resamples whole top-level clusters, recomputes the selected within- and between-cluster correlations, and reports percentile bootstrap confidence intervals. This keeps the package's descriptive estimands while avoiding row-level independence assumptions.

Inspired by the `psych::statsBy` function, `wbCorr` allows you to calculate, extract, and plot within- and between-cluster correlations for further analysis.

Value

A `wbCorr` object that contains within- and between-cluster statistics. Use the `get_table()` function on the `wbCorr` object to retrieve a list of the full correlation tables. Use the `summary()` or `get_matrix()` function on the `wbCorr` object to retrieve various correlation matrices, including ICCs in the merged ones. Use `get_ICC()` in order to get all intra class correlations ($ICC(1,1)$). Finally, use `to_excel()` on a table or matrix (or list of matrices) to save them.

See Also

[get_table](#), [summary](#), [get_ICC](#), [plot](#), [to_excel](#)

Examples

```
# importing our simulated example dataset with pre-specified within- and between- correlations
data("simdat_intensive_longitudinal")

# create a wbCorr object:
```

```

correlations <- wbCorr(simdat_intensive_longitudinal,
                      'participantID')

# optionally compute sample-size weighted between-cluster correlations:
weighted_correlations <- wbCorr(simdat_intensive_longitudinal,
                                'participantID',
                                between_weighting = 'cluster_size')

# quick cluster-bootstrap example; use more bootstrap samples in applied work:

bootstrapped_correlations <- wbCorr(simdat_intensive_longitudinal,
                                    'participantID',
                                    inference = 'cluster_bootstrap',
                                    nboot = 20)

# optionally estimate cluster means from all rows available for each variable:
all_available_correlations <- wbCorr(simdat_intensive_longitudinal,
                                    'participantID',
                                    centering_rows = 'all_available')

# returns a list with full detailed tables of the correlations:
tables <- get_table(correlations) # the get_tables() function is equivalent
print(tables)

# returns a correlation matrix with stars for p-values:
matrices <- summary(correlations) # the get_matrix() and get_matrices() functions are equivalent
print(matrices)

# Plot the centered variables against each other
plot(correlations, 'within')
plot(correlations, which = 'b')

# Store the list of correlation matrices to excel
to_excel(matrices, path = tempfile(fileext = ".xlsx"))

```

 wbCorr-class

wbCorr Class

Description

A class representing within- and between-cluster correlations.

Details

The `wbCorr` class is used to store within- and between-cluster correlations and provides methods for printing and summarizing the correlations.

See Also

[wbCorr](#)

Index

`get_ICC`, [2](#), [13](#)
`get_icc` (`get_ICC`), [2](#)
`get_ICCs` (`get_ICC`), [2](#)
`get_matrices` (`get_matrix`), [3](#)
`get_matrix`, [3](#), [10](#)
`get_table`, [4](#), [13](#)
`get_tables`, [4](#), [10](#)
`get_tables` (`get_table`), [4](#)

`plot`, [13](#)
`plot`, `wbCorr`-method, [5](#)
`plot.wbCorr` (`plot`, `wbCorr`-method), [5](#)
`print`, `wbCorr`-method, [7](#)
`print.wbCorr`, [8](#)
`print.wbCorr` (`print`, `wbCorr`-method), [7](#)

`show`, `wbCorr`-method, [8](#)
`show.wbCorr` (`show`, `wbCorr`-method), [8](#)
`simdat_intensive_longitudinal`, [8](#)
`summary`, [5](#), [13](#)
`summary`, `wbCorr`-method (`get_matrix`), [3](#)
`summary.wbCorr` (`get_matrix`), [3](#)

`to_excel`, [9](#), [13](#)

`update_wbCorr`, [10](#)

`wbCorr`, [3–8](#), [10](#), [11](#), [15](#)
`wbcorr` (`wbCorr`), [11](#)
`wbCorr`-class, [14](#)